Gerenc.tecnol.inform. Vol 6|No 16|13-30,2007

EXTENSIÓN DE UN DATA WAREHOUSE USANDO ETIQUETAS LINGÜÍSTICAS Y OPERADOR CUBE

AUTOR
Angélica Urrutia Sepúlveda
Doctora
Universidad Católica el Maule
aurrutia@ucm.cl
CHILE

AUTOR
Diego Egaña
Magíster©
Universidad Santiago de Chile
degaña@gmail.com
CHILE

Fecha de Recepción : 9 de Septiembre de 2007
Fecha de Aceptación : 11 de Noviembre de 2007
ARTICULO TIPO 1

RESUMEN.

La motivación de este trabajo, es poder extender un Almacén de datos o Datawarehouse que entrega información cuantitativa a información cualitativa, lo que permite darle un significado lingüístico a la información numérica que requiere la gestión de la información en la toma de decisiones. Para extender el Datawarehouse, se utiliza la lógica difusa, teoría matemática que permite la especificación de requerimientos graduales o flexibles. Esto flexibiliza la representación y manipulación de información de los datos en una Base de Datos, donde los requerimientos de los usuarios son de una naturaleza de la aplicación de datos imprecisos. La solución que se propone, es desarrollar un conjunto de pasos que denominamos MFDW y parten desde la base de datos fuentes hasta la implementación de cubos extendidos con etiquetas lingüísticas (FuzzyMedida). Para realizar esta extensión, se crea M-FIRST que es una extensión del FSQL [5] para base de datos relacional difusas, y se anexa a un diseño estrella o copo de nieve de modelos multidimensionales para un determinado almacén de datos. Aquí se toma una medida en particular del Datawarehouse, se le agregan etiquetas lingüísticas requeridas por el usuario, las cuales varían de acuerdo al cruce de dimensiones que se realicen sobre esa medida. Finalmente se construye una herramienta Front-End (Web), que permite almacenar y consultar información imprecisa.

PALABRAS CLAVE

Gestión de información
Bases de datos multidimensionales
Almacén de datos difusos
Medidas imprecisas.

ABSTRACT

The purpose of this investigation is to extend the generally accepted concept of datawarehousing that blends ordinal quantitative and qualitative information, to include class descriptors that describe a spectrum of values for sets of domain dependent data. It is proposed that this kind of `fuzzy`data, when combined with the

ordinal data, will enrich the information provided to managers by making it more meaningful. Fuzzy logic is used to define this broad spectrum data because it enables formal requirement specifications to be generated. This approach introduces an element of both flexibility and precision for the description, representation and manipulation of the data not usually available in previous methods. The concept proposed here is implemented by developing a series of steps referred to as MFDW that begins with describing the source data base and finishes with the implementation of extended cubes that have associated linguistic semantic labels (FuzzyMedida). To perform this operation, it creates a processing platform framework called M-FIRST, which is an extension of FSQL [5], which is used for fuzzy relational data base design. This then enables a star schema design to be generated, which is a multidimensional model for the datawarehouse. In this paper we take an example datawarehouse, add linguistics labels required by the user to inform the fuzzy modeling tool and finally we build a Front-End tool (web) that allows for the storage and enquiry of imprecise information that can be blended with the ordinal data already in the pre-formed data set.

KEYWORDS

Online Analisys Proccess
Multidimensional model
Fuzzy Datawerehouse
Fuzzy Data

1. INTRODUCCIÓN

Al menos dos tipos diferentes de aplicaciones computacionales pueden ser distinguidas en el manejo de información de una compañía: procesamiento de transacciones en línea OLTP (Online Transaction Proccesing) y procesamiento analítico en línea OLAP (Online Analisys Proccess). El primero soporta el proceso primario de la compañía con sistemas de procesamiento transaccionales y el segundo le concierne el manejo de la información de gestión que se obtiene a partir del procesamiento primario registrado en las aplicaciones OLTP [11]. También, se considera que por lo general, las aplicaciones de una compañía tienen sus datos en alguna herramienta que permita modelos relacionales para sus bases de datos. El modelo de datos relacional fue introducido por Codd, y durante 1990, surgen las herramientas de bases de datos que soportan modelo de datos relacional (Oracle, SQL Server, ...), además últimamente se tienen herramientas de bases de datos que soportan modelo de datos multidimensional en plataformas relacionales (Análisis Server, Oracle Express, ...), usadas cuando el objetivo es analizar los datos (OLAP), por sobre la ejecución de transacciones en línea (OLTP).

Los Data Warehouse son conocidos como un proceso de integración y colección de datos que han sido registrados en el tiempo en bases de datos operacionales, mas bien llamados datos fuentes, y es principalmente usado en la toma de decisiones estratégicas por medio del procesamiento analítico en línea (OLAP). Es esencialmente una base de datos que integra información, a menudo histórica, pudiendo agregar información extraída desde bases de datos múltiples, heterogéneas, autónomas y distribuidas fuentes de información.

Su principal objetivo es consolidar información proveniente de las bases de datos fuentes, y hacerla disponible para el análisis de información a nivel gerencial para la toma de decisiones. Dichas bases de datos son construidas con la información que registran las transacciones de los negocios de la organización. Los Datawarehouses, entregan información a nivel cuantitativo (Ej. ¿Cuánto vendió la sucursal x del producto y para el período z?), existiendo diferentes definiciones de Datawarehouse según el autor que se consulte, la más utilizada por su sencillez, claridad y fácil comprensión es la entregada por [10]:

"Un Datawarehouse es una colección de datos integrados, temáticos, no volátiles variantes en el tiempo, organizados para soportar necesidades empresariales orientadas a la toma de decisiones".

Se puede concluir, que un Datawarehouse, es el proceso de extraer y filtrar datos de las operaciones comunes de las empresas, procedentes de los distintos subsistemas operacionales, para transformarlos, integrarlos, totalizarlos y almacenarlos en un depósito o almacén de datos, para poder acceder a ellos cada vez que se necesite mediante mecanismos flexibles para el usuario.

1.1 ¿Que se pretende?

Generar un conjunto de pasos que permitan construir un DataWarehouse (DW), cuya información consultada en web, apoyen al usuario de gestión de la información, cuando los datos cuantitativos no le son eficientes, aportando resultados en términos cuantitativos. Por ejemplo, la consulta que da el comportamiento de las ventas de un producto en malas, regular o buenas, pueden ser conceptos muy útiles para la toma de decisiones de marketing, pero para tales requerimientos de negocios la consulta debería ser capaz de proveer respuestas a consultas en términos lingüísticos de: mala, regular o buenas, con sus correspondientes valores asociados.

Por tanto, aquí se presenta una propuesta para extender un Datawarehouse cuantitativo a un Datawarehouse cuantitativo-cualitativo, generando tres capas o niveles

y 11 pasos para su construcción (véase Figura 7). Nuestra propuesta se aplica a un caso práctico de una empresa del rubro comercio exterior, con una base de datos fuentes en SQL Server 2000, donde, la generación del DW cuantitativo es en la herramienta Análisis Manager del SQL Server versión 2000. Finalmente, para la extensión del DW cualitativo, se utiliza la base de datos relaciones difusa FSQL [5] y se crea M-FIRST. Cabe destacar que el resultado de la investigación presentada aquí, fue producto de una tesis del Magíster en Tecnologías de la Información de la Universidad de Santiago de Chile.

Los apartados que se presentan en este trabajo son: conceptos básicos, discusión bibliográfica, pasos para la implementación de un DW difuso, conclusiones, trabajos futuros y referencias bibliográficas.

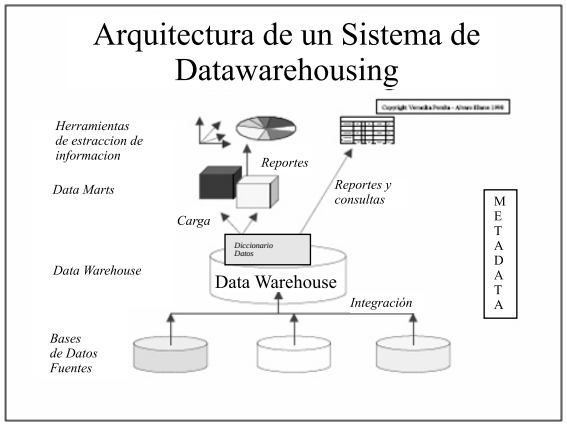
2. CONCEPTOS BÁSICOS

En este apartado se presenta, una breve descripción de algunos conceptos que se utilizaron en la investigación.

2.1 Componentes del Datawarehouse

Un DataWarehouse, es una base de datos con resúmenes precalculados que centraliza una parte o todos los datos de una empresa. Esta base de datos se carga a partir de los datos existentes, en una o varias bases de datos fuente, con un proceso de integración que consiste principalmente, en la extracción, limpieza y resúmenes de datos que sean relevantes para la toma de decisiones por períodos predeterminados. En la Figura 1, se presenta la arquitectura típica de una DataWarehouse.

Figura 1: Arquitectura de un Sistema de Datawarehousing.



A partir del DW, que contiene la metadata extraída de la fuente, se construyen Data Marts o también llamados cubos, orientados a un aspecto o área específica de la empresa. (Por Ej., puede existir un cubo para ventas, otro para gastos, ...). Con el DW o con un Data Marts, se pueden construir reportes y gráficos utilizando herramientas de apoyo, la más común es la planilla de cálculo. Las características del almacenamiento de datos del cubo o Data Mart es disponer de los datos de análisis para realizar consultas de gestión. Su característica principal es la realización de un conjunto de cálculos y almacenamiento, para que las consultas se lleven a cabo en el menor tiempo posible. En la actualidad existen 3 formas de trabajar estos procesos OLAP:

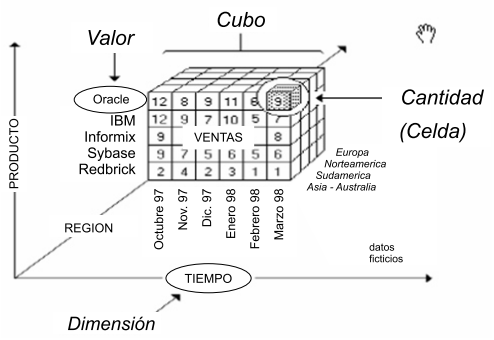
- OLAP Multidimensional (MOLAP): Este método de almacenamiento es una estructura de datos multidimensional o hipercubo de rendimiento y almacenamiento óptimo y sus componentes son dimensiones y medidas que se almacenan de forma independiente a los datos fuentes.
- OLAP Relacional (ROLAP): En este modo, los datos tanto como los valores calculados, no se almacenan en una estructura multidimensional, sino que se consultan en una base de datos relacional y generar vistas con instrucciones SQL de cubos, entre otras.
- OLAP Híbrido (HOLAP): Es una solución intermedia entre MOLAP y ROLAP combinándolas. Este tipo de almacenamiento, mantiene consultas en la base de datos relacional y las agregaciones en un almacén MOLAP.

Modelos Multidimensionales: La base de datos multidimensional es un tipo de estructura que permite consultas más complejas, donde la información se representa como matrices multidimensionales, cada matriz se denomina Data Marts o Cubo (hipercubo).

A los ejes de un cubo se les llama dimensiones, en la Figura 2, las dimensiones son: Producto, Tiempo y Región. Al dato que se presenta en la matriz se le llama medidas, en la Figura 2, medidas son las Ventas. Una celda, es la posición formada por la intersección de cada uno de los elementos de las dimensiones que forman el cubo, donde una celda puede contener una o más medidas. En el caso particular de la celda marcada con el valor 9 (extremo superior derecho del cubo de la Figura 2), significa que hubo un total de 9 ventas (medida venta) de productos Oracle (dimensión producto), en Marzo de 1998 (dimensión tiempo), en la región Asia-Australia (dimensión región).

El valor contenido en la celda, por si solo no indica nada, sin embargo para el usuario ese valor tiene un significado lingüístico asociado (Por Ej., fue buena o mala venta).

Figura 2: Ejemplo de una estructura básica de un cubo.



La principal característica del modelado de datos multidimensional es la división de los datos en hechos (conjunto de medidas) y dimensiones [9].

Diseño Lógico Multidimensional: La estructura básica para un modelo multidimensional definida por dos elementos: esquemas y tablas, existiendo dos tipos básicos de tablas: Las de hecho (Fact) que contienen valores de las medidas de negocios y las dimensionales que contienen el detalle de los datos que se encuentran asociados a las tablas de hecho. Los esquemas corresponden a la colección de tablas que conforman el diseño, son:

1.Esquema Estrella: Su estructura básica es una tabla central y un conjunto de tablas relacionadas con la tabla central, de ahí su nombre, ya que es una representación en forma de estrella. El centro de la estrella consiste en una o varias medidas (hechos) y las puntas de la estrella son las tablas dimensionales. Si toda la información se concentra en una tabla central, lleva el nombre de tabla Fact o Hecho.

2.Esquema Copo de Nieve: La diferencia con el esquema anterior, es la estructura de tablas dimensionales, que en el modelo copo de nieve, están normalizadas, ya que, las dimensiones se estructuran en jerarquías de agregación y cada jerarquía es una tabla asociada a su dimensión en forma jerárquica.

Operaciones Multidimensionales: La operaciones multidimensionales son sobre los cubos y se pueden agrupar en tres conjuntos básicos:

De selección y visualización Slice & Dice: tiene tres operaciones asociadas: a) selecciona "dimensiones de trabajo" de un cubo mayor (Slice), b) selecciona "secciones" del cubo en función de valores de las dimensiones (Dice o Filtrado) y c) que permite "presentar" diferentes planos de un cubo (Rotación).
De Agregación: está constituido por operaciones que surgen de realizar "movimientos" en las jerarquías de las dimensiones. Cuando se "sube" de nivel por una jerarquía, se agrupan todos los valores del nivel original que están relacionados con el mismo valor del nivel superior, mientras que al "bajar" por la jerarquía se produce la desagregación de dichos valores. La primera operación se conoce como DrillUp y la segunda, su inversa, como DrillDown. Cuando se realiza un DrillUp, se debe calcular una nueva medida en función del conjunto de los valores de las medidas que se agrupan, a esta operación se le llama Roll-up o Consolidación (típicamente funciones de agregación de SQL sum, avg, etc.).
De Relacionamiento: partir de un cubo se puede acceder a otros datos. Si éstos últimos están en un cubo, la operación se suele llamar de Drill-Across, mientras que si están en el Data Warehouse o en la base operacional, la operación se suele llamar Drill-Through. Dado un cubo, al aplicar operaciones de DrillUp o DrillDown, se recorre un espacio de cubos". Dicho espacio está determinado por las dimensiones que participan en el cubo origen y la forma en que se deben realizar los cálculos con las medidas (RollUp) en cada DrillUp.

2.2 Modelo conceptual para Datawarehouse

En la actualidad no existen modelos conceptúales estándares para recoger la especificación de requerimientos de un Data Warehouse, nosotros hemos estudiado a (Golfareli et al., 1999) que parte de un modelo relacional y desde ahí genera el diseño, siendo su desventaja el que se debe conocer el modelo de la base de datos fuentes. Otra representación es CMDM, que es un modelo para representar los requerimientos de un sistema de toma dediciones, sin ser necesario conocer la base de datos fuente. CMDM se basa en conceptos de modelo multidimensional e independiente de la implementación, permitiendo modelar esquemas multidimensionales a partir de los requerimientos de gestión [8].

Estructuras en CMDM: El objetivo fundamental de CMDM, es permitir la especificación de una determinada realidad en términos multidimensionales, que a partir de tres componentes, permite generar las especificaciones de un esquema en CMDM, explicadas a continuación:

Niveles: Un nivel representa un conjunto de objetos que son de un mismo tipo y cada nivel debe tener un nombre y un tipo. Para representar el esquema de un nivel se utiliza un rectángulo que contiene el nombre y las jerarquías de ese tipo de nivel (véase Figura 8, D-tiempo).
Dimensiones: Una dimensión está determinada por una o varias jerarquía de niveles. De esta forma, el esquema de una dimensión está representado por un rectángulo dentro del cual aparece el nombre de la

dimensión y un grafo dirigido en donde los nodos son los niveles que participan en esa dimensión. De igual forma se representan las medidas (véase Figura 8).

Relaciones Dimensionales: Una relación dimensional representa la unión del conjunto de uno o más cubos que se pueden construir a partir de los niveles de un conjunto de dimensiones y medidas. Se asume que en cada uno de los cubos que pertenecen a la instancia de la relación dimensional, debe aparecer al menos un nivel de cada una de las dimensiones que participan en la relación (véase Figura 9).

En CMDM, un cubo es una función de producto cartesiano de las instancias de los niveles, de esta forma, cualquier nivel puede cumplir el rol de medida. Por lo tanto, el esquema de una relación dimensional está dado por un grafo en forma de estrella. El nodo central es de forma oval y tiene el nombre de la relación dimensional y los nodos "satélite" de forma rectangular y tienen el nombre de cada una de las dimensiones que participan en la relación. Un ejemplo de aplicación de CMDM y su mapeo a cubo para DW se encuentra en [16].

2.3 SQL Server y Olap

Existen varias herramientas que permiten gestionar la información en OLAP, entre ellos esta un producto Microsoft que incorpora cubos multidimensionales por medio de Análisis Manager desde el SQL Server 7, pasando por el SQL Server 8 o 2000 y actualmente con el SQL Server 2005. Una vez instalado el servidor OLAP, permite realizar la construcción de un Datawarehouse con relativa facilidad [3,17].

Características del Analysis Services: Para construir un Datawarehouse, Análysis Services provee asistentes, editores, herramientas e información incluida con la herramienta Analizáis Manager. La consola de aplicación provee una interfase para acceder al manejo de cubos y datos de este. Incluye un asistente de cubos, el cual permite construir toda la estructura necesaria para crear un Cubo OLAP. El asistente guía a través del proceso completo del diseño del cubo y proceso de implementación, desde la extracción de los datos fuentes hasta la creación de dimensiones y medidas. Con una simple operación drag-and-drop, se puede editar la estructura de un cubo, modificar y creando una nueva con el asistente de cubos. Usando el asistente de dimensiones, se pueden de crear de manera fácil estructuras de dimensiones (véase punto 4.7 y Figura 10).

En general SQL Server 2000 Analysis Manager, provee una interfaz gráfica para la creación y mantenimiento de Cubos OLAP. Las operaciones cubos (Slice, Dice, Roll-up, etc), se pueden efectuar de manera fácil por medio del uso del mouse, para realizar una operación Dice, solo basta elegir el valor de una dimensión en particular. Si las dimensiones están jerarquizadas, realiza de manera automática el Drill-down o Drill-up, según corresponda.

2.4 Conjuntos Difusos

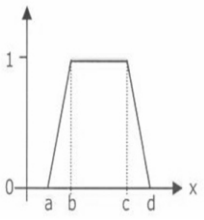
La teoría de conjuntos difusos propuesta por Zadeh [18], nos permite representar de forma adecuada la información difusa (incierta) presente en las bases de datos, basándose en la idea de que existen conjuntos en los que no esta claramente determinado si un elemento pertenece o no al conjunto. Por ejemplo, el conjunto de las personas que son "altas" es un conjunto difuso, pues no está claro el límite de altura que establece, y a partir de que medida una persona es alta o no lo es. Ese límite es difuso y, por tanto, el conjunto que delimita también lo será. La definición esta dada por, un conjunto difuso A sobre un universo de discurso *U* es un conjunto dado por:

A = {μ_A (u)/u : u pertenece a U, u_A (u) pertenece al intervalo [0,1}, donde, ì es la llamada función de pertenencia y μ_A (u) es el grado de pertenencia del elemento u al conjunto difuso A. Este grado oscila entre los extremos 0 y 1, μ_A (u)=0, indica que u no pertenece en absoluto al conjunto difuso A, μ_A (u)=1 indica que u pertenece totalmente al conjunto difuso A.

Si el grado de pertenencia se encuentra en el conjunto {0,1}, entonces, el conjunto que genera no es difuso, llamado "crisp", tradicional o preciso. En el caso en que se desconozca la información o no se está seguro, el valor no puede ser aplicado, por lo tanto, la lógica difusa permite almacenar este tipo de datos, ya que acepta valores intermedios, indefinidos y nulos. La gráfica asociada a la lógica difusa es la siguiente Figura 3.

Figura 3: Distribución Trapezoidal para conjuntos difusos.

$$T(x) = \begin{cases} 0 & \text{si } (x \le a) \text{ o } (x \ge d) \\ (x-a)/(b-a) & \text{si } x \in (a,b] \\ 1 & \text{si } x \in (b,c) \\ (d-x)/(d-c) & \text{si } x \in (b,d) \end{cases}$$



La distribución trapezoidal puede tomar variadas formas dependiendo si a, b, c, d son de igual o diferente valor. Por ejemplo, si b y c tienen el mismo valor la figura asociada es un triangulo representando valores aproximados. A esta función se le puede asociar una etiqueta lingüísticas. Otros ejemplos están en [6].

3. DISCUSIÓN BIBLIOGRÁFICA

Este apartado presenta algunos trabajos que sirven de apoyo y punto de partida sobre la temática de extensiones de Datawarehouse y de componentes difusas.

3.1 P. Kumar, R. Krishna, S. Kumar5 [12]

En el trabajo titulado "Fuzzy OLAP Cube for Qualitative Analysis", los autores presentan como construir un cubo difuso OLAP para realizar análisis cualitativo sobre un Datawarehouse. Presentan cinco operaciones OLAP que pueden ser realizadas en un cubo difuso, los autores definen una Dimensión Difusa como la dimensión en la cual los atributos que están involucrados en el análisis pueden ser fuzzificados. El valor del atributo en la dimensión difusa (FD), son mapeados a un conjunto de atributos con dos valores, conteniendo el término lingüístico y su correspondiente valor de pertenencia definida como:

F(FD) = dominio (FD) » <L, [0,1]> , Donde L es el valor lingüístico.

Un cubo OLAP difuso es un cubo de datos en el cual la medida y una o mas dimensiones son fuzzificadas. Un cubo OLAP difuso N-dimensional, es definido como una función de mapeo

F (FC): dominio (FD $_1$) x dominio (FD 2) x …x dominio(FD $_n$) » {L, [0,1]}

En este trabajo se aplica el algoritmo de agrupamiento CLARANS, sobre cada atributo para encontrar los conjuntos difusos a agrupar, este algoritmo considera el agrupamiento de datos buscando la similitud entre ellos, se construyen K particiones (correspondientes a las etiquetas lingüísticas a utilizar) en donde a lo menos existe un elemento. El algoritmo iterará hasta un criterio definido para su detención, después de la fuzzificación, los k agrupamientos se transforman en K términos lingüísticos para cada atributo. CLARANS, en la búsqueda de las k agrupaciones para cada atributo, considera otros atributos que influyen en la representación lingüística del atributo original, lo cual ayuda a obtener el análisis cualitativo del cubo difuso OLAP.

A modo de ejemplo, los autores consideran tres etiquetas lingüísticas: buenas, promedio, malas, para la medida ventas, de acuerdo a esto, el algoritmo CLARANS seleccionó tres agrupaciones que representan el dominio de cada una de las tres etiquetas lingüísticas. En el trabajo, se realizó la consulta de las ventas con la localización noroeste, y edad niños, para el año 2002. El resultado fue una venta promedio de 7374, la misma consulta pero haciendo Roll-up a la dimensión localización a norte dio como resultado 26710, lo que es considerado, como buena venta. La implementación de este trabajo fue en Análisis Server de SQL Server.

3.2 L. Fen, Tharams. Dillon [13]

En el trabajo de nombre, "Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses", los autores proponen que para un usuario experto del dominio de datos de algún DataWarehouse, la semántica conlleva a etiquetas lingüísticas, que son más naturales, significativas y conocidas. En efecto, mucho del razonamiento humano en la vida real involucra el uso de términos lingüísticos por sobre números rígidos. El manejo de estos términos lingüísticos, necesariamente involucra agregar una interpretación entre el usuario y los desarrolladores que trabajan con la información contenida en bases de datos, con el fin de agregar funciones de pertenencias. Para agregar representación semántica a un DataWarehouse, los autores propone un modelo de tres capas llamados: nivel cuantitativo (numérico) de totalización, nivel cualitativo (categoría) de totalización y nivel cuantificador de totalización.

Nivel 1 Cuantitativo: Este nivel esta compuesto por varias vistas agregadas en el DataWarehouse tradicional. Dado que la vista de datos es a menudo calculada y derivada por funciones numéricas de agregación tales como: sum, avg, count, min, max, etc. Los autores llaman a este nivel como el nivel cuantitativo (numérico). Este nivel representa a la construcción del DataWarehouse tradicional sin extensión difusa.

Nivel 2 Cualitativo: Considerando el hecho que la percepción humana y de pensamiento están generalmente basada sobre etiquetas lingüísticas en vez de un número en particular, Los autores, al dato numérico del nivel 1 lo asocia a un concepto descriptivo y categoría, de modo que la decisión a tomar a cerca de un problema este representada en el Datawarehouse. Aquí se aplica la técnica de conjuntos difusos para facilitar este proceso. Por ejemplo, de acuerdo al total de ventas de un producto en el nivel 1, se pueden categorizar estas en buenas (regular o malas) ventas, sobre el nivel 2, usando ciertas funciones de pertenencias. Luego, el usuario puede directamente involucrar estos términos lingüísticos sobre una consulta del Datawarehouse. Como un atributo de medida, puede ser usualmente asociado con muchos conceptos, (Por ej., las ventas pueden ser descritas como muy buenas, malas, buenas, medias, muy malas, etc.) un operador FUZZ-TERM es introducido para que el administrador del Datawarehouse, indicando un conjunto base de términos lingüísticos que sean interesantes e importantes para el proceso de toma de decisiones.

FUZZ-TERM: <conjunto de términos lingüísticos> <conjunto de funciones miembros> ON <atributo medible> FROM <tabla>

Nivel 3 Cuantificativo: En adición al valor entregado y categorización del total de cada valor de medida individual, a menudo, los usuarios conocedores del negocio, desea tener cierto conocimiento más preciso dentro del funcionamiento total del valor de un grupo de medidas.

Lo relevante en el trabajo de estos autores, es la definición de los operadores FUZZ-TERM y FUZZ-QUANTIFIER, ambos se aplican sobre diferentes vistas que representa cubos en diferentes niveles, ambos incluyen representación lógica difusa y estructuras. Lo importante, es que el administrador del Datawarehouse conozca el dominio de los datos para poder fabricar cubos que den soporte a sus actividades de negocios. Un problema que se puede dar, es definir la función de miembro apropiada para medida (nivel cualitativo) o dimensión (nivel cuantitativo).

En la implementación de este trabajo se considera el software Warehouse Manager, es el responsable del almacenado, manejo y mantención de los datos en el DataWarehouse. En las situaciones que requieren aplicaciones aproximadas al razonamiento humano y vistas preceptúales el administrador del DataWarehouse instruirá al Warehouse Manager, el cual procederá a desarrollar la semántica para el DataWarehouse a través del siguiente proceso: La etapa de construcción de la semántica, la Metadata es un elemento esencial en el Sistema de Datawarehouse. La semántica del DataWarehouse es modelada a través de los tres diferentes niveles; cuantitativo, cualitativo y cuantificativo.

3.3 P. Cheung, R. Lau, A. Lee, L. Tsoi and F. Yip [14]

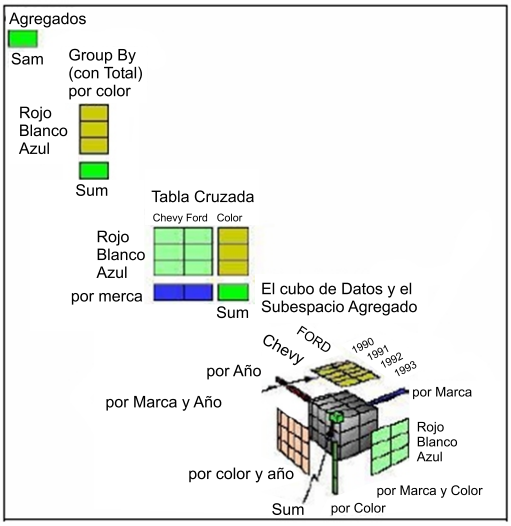
Aquí se discuten y presentan una propuesta del operador CUBE considerado en el estándar del SQL2003 (Melton 2003). Es una función de análisis de datos, que forma parte de la instrucción Group By y permite generar datos para la gestión de la información.

En el trabajo "Data Warehouse and OLAP" los autores discuten que, el análisis de requerimientos de datos, es extraído desde los datos relevantes de las bases de datos fuentes, a los cuales se les agregan campos totalizados, y que generalmente ocupan una gran cantidad de datos. La extracción de datos y agregación es común en sentencias SQL, por lo general utilizan las funciones de agregación de datos, Count(), Sum(), Min(), Max(), y Avg() y últimamente operadores CUBE y Rollup. Para agrupar los resultados se usa la instrucción Group By.

En su trabajo los autores plantean algunos problemas con el Group By y proponen el operador CUBE, donde el operador relacional Group By, divide una tabla dentro de grupos y cada grupo se agrega con una función. Esta función de agregación resume algunas columnas del grupo, retornando un valor para cada grupo, tal como se muestra en la Figura 4 a). En cambio el operador CUBE, construye una tabla con todos los valores agregados y es un operador relacional, tal como se muestra en la Figura 4 y 5. Su definición es:

v1, v2,…,vn, f()
v1, v2, … , 'Todos', f()
…
v1, 'Todos', … , 'Todos', f()
'Todos', 'Todos', …, 'Todos', f() , donde f() es una función de agregación.

Figura 4: Operador Group By y Operador CUBE.



Un ejemplo de la sintaxis del operador CUBE, sería:

Figura 5: a) Sintaxis operador CUBE, b) Ejemplo función Grouping en CUBE.

```
SELECT Modelo, Año, Color, SUM(ventas),
                           GROUPING(Modelo),
                           GROUPING(Año),
                           GROUPING(Color)
FROM Ventas
GROUP BY CUBE(Modelo, Año, Color);
```

El operador CUBE, es la generalización en N-dimensiones de funciones de agregación simples. El cubo de datos de 0D es un punto, el de 1D es una línea con un punto, el de 2D es una tabla cruzada, un plano, dos líneas y un punto y el de 3D es un cubo con tres

intersecciones de tablas cruzadas en 2D, como se representa en la Figura 4.

La suma global puede ser la tupla: {Null, Null, Null, 941, True, True, True}. Está expresión muestra que cada valor Null representa el valor "Todos", en el caso que fuera Falsa, representaría un valor nulo. El cubo de datos se basa en la idea de usar los valores "Todos" para el Group By y las agregaciones. El operador CUBE, es un operador relacional para simplificar la agregación, generalizar agregaciones y generar tablas cruzadas.
Los autores proponen que ciertas formas de análisis de datos son difíciles sino imposibles con los constructores SQL y formulan tres problemas recurrentes del operador Group By: Histogramas, Totales Roll-Up y sub-totales para drill down de tablas cruzadas, discutidas en su trabajo, Otros autores que discuten este tema son [13].

Operador CUBE en SQL Server [15]: El operador CUBE genera un conjunto de resultados que forman un cubo multidimensional. Un cubo multidimensional es una expansión de datos de hechos o datos que registran eventos individuales. La expansión se basa en columnas que el usuario desea analizar, estas columnas se llaman dimensiones. El cubo es un conjunto de resultados que contiene una tabla cruzada de todas las combinaciones de dimensiones posibles [3].

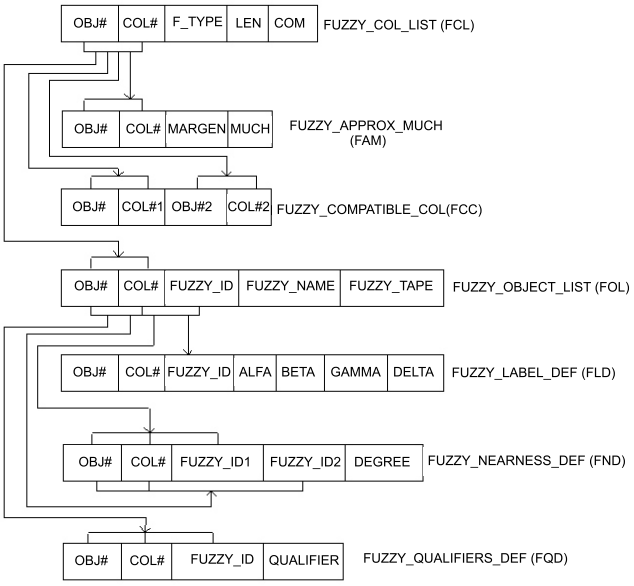
El operador CUBE se especifica en la cláusula GROUP BY de una instrucción SELECT. La lista de selección contiene las columnas de dimensión y las expresiones de funciones de agregado. GROUP BY especifica las columnas de dimensión y las palabras clave WITH CUBE. El conjunto de resultados contiene todas las combinaciones posibles de los valores de las columnas de dimensiones, junto con los valores de agregado de las filas subyacentes que coinciden con esa combinación de valores de dimensión.

Los administradores de bases de datos BD2 de IBM [4] y Oracle Express, entre otros, ejecutan consultas OLAP con el operador CUBE.

3.4 Propuesta del FSQL Y FIRST (Fuzzy Interface for Relational SysTems)

La propuesta de bases de datos difusas une, la teoría relacional de bases de datos con la teoría de conjuntos difusos, permitiendo el almacenamiento de información difusa para el tratamiento y consulta de la información imprecisa. La Figura 6, muestra una extensión de componentes difusas para administradores de bases de datos relacionales que es levantada mediante un procedimiento y permite almacenar, etiquetas lingüísticas, valores de etiquetas lingüísticas para datos ordenados y no ordenados, cuantificadores etc.

Figura 6: FIRST propuesta en [5] para FSQL.



En [6] se proponen ocho tipos de datos para ser representados con la teoría de conjuntos difusos, a partir de tres tipos de atributos difusos que son los propuestos en [5] susceptibles de tratamiento impreciso. Estos datos se clasifican según el tipo del dominio que les subyace (continuo o similitud) y almacenamiento información imprecisa. Estos son:

Tipo 1: Atributos que son tradicionales crisp, sin imprecisión, pero también admiten que en su dominio se pueda definir alguna etiqueta lingüística para usar en consultas.
Tipo 2: Atributos que admiten tanto datos con o sin imprecisión en forma de distribución de posibilidad sobre un dominio subyacente ordenado. Además permite la representación de datos de tipo Unknown, Undefined y Null.
Tipo 3: Atributos que definen algunas etiquetas que son escalares con una relación de similitud definida sobre ellas, de forma que esta relación indique en qué medida se parecen entre sí cada par de etiquetas, también permite la representar datos e tipo Unknown, Undefined y Null.

Representación de Imprecisión en la Base de Datos SGBD. Cada tipo de datos, descrito anteriormente, tiene su respectiva representación en FSQL con información asociada en la FMB (Base de Metaconocimiento Difuso) y FIRST:

a)Atributos Difusos Tipo 1: Este tipo de atributos se representa igual que los datos precisos, pudiendo ser asociados a etiquetas lingüísticas en

consultas, por lo tanto, son atributos clásicos que admiten el tratamiento difuso.

b)Atributos Difusos Tipo 2: Son atributos que pueden recoger "atributos sobre referencial ordenado" asociados a una etiqueta lingüística para su almacenamiento como para consultas.

Así, vemos que un atributo difuso Tipo 2, está compuesto por 5 atributos clásicos, insertando 5 columnas por cada dato impreciso en la base de datos con la nomenclatura FT, F1, F2, F3, y F4 descritos a continuación:

FT: Almacena el tipo de valor que corresponde al dato que queremos almacenar, indicando su representación. Según lo visto, puede ser: Unknow (0), Undefined (1), Null (2), Crisp (3), Label (4), Intervalo (5), Aproximadamente (6) o Trapezoidal(7).

- **F1, F2, F3 y F4:** Atributos cuyo nombre se forma añadiendo los números 1, 2, 3 y 4 al nombre del atributo almacenando la descripción de los parámetros que definen el dato y que depende del tipo de valor (FT) al que pertenezca:
- **Unknow, Undefined y Null:** Estos 3 valores no necesitan ningún parámetro, por lo que todos ellos permanecen a NULL (entendiendo este valor como el NULL del SGBD anfitrión.
- **Crisp:** Un valor de tipo preciso, necesita tan solo un parámetro, F1, en el cual se almacenará el valor normal en cuestión.
- **Label:** Igualmente, un valor de tipo etiqueta solo necesita un parámetro para almacenar el identificador asociado a dicha etiqueta (Fuzzy_Id). Ese indicador es útil para poder acceder a la FMB y obtener la descripción asociada a esta etiqueta.
- **Intervalo:** Necesita los dos valores extremos del intervalo [n, m], que son almacenados en F1 y F4.
- **Aproximadamente:** Este valor solo necesita un valor que se almacena en F1 y que es el valor central de la distribución de posibilidad triangular, d. Sin embargo, para reducir operaciones (tanto matemáticas como de acceso a datos), se aprovechan los atributos F2, F3 y F4 para almacenar los valores d - margen, d + margen y margen, respectivamente.
- **Trapecio:** Necesita forzosamente almacenar los 4 valores que identifican a un trapecio: $[\alpha, \beta, \gamma, \delta]$.
- **Atributos Difusos Tipo 3:** Son atributos sobre "dominio discreto no ordenado". Estos atributos recogen datos escalares simples (Simple) o distribuciones de posibilidad (Distr.Pos.) sobre dominios escalares.
- **FT:** El tipo de valor que corresponde al dato que queremos almacenar. Este puede ser: Unknow (0), Undefined (1), Null (2), Simple (3) o Distribución de Posibilidad (4). Lista de n parejas, con n >= l, del tipo (*valor de posibilidad, etiqueta*), (FPl, Fl)... (FPn, Fn): En estos atributos se almacenan los datos de la distribución de posibilidad que deseamos almacenar. En un valor de tipo Simple solo se usa la primera pareja y el valor de posibilidad debería ser 1 (para estar normalizada).

En la extensión del DW difuso que mostramos a continuación, utilizamos el tipo de datos 1 y 2, el tipo de datos 3 se deja para trabajo futuro.

4. PASOS PARA LA IMPLEMENTACIÓN DE UN DW DIFUSO

En las etapas de desarrollo para la construcción del DataWarehouse Difuso, al que hemos llamado MFDW (Method Fuzzy DataWarehouse), considera un conjunto de 11 pasos, tal como lo muestra la Figura 7, asociados a tres Fases fundamentales: la primera es la especificación de requerimientos de la información de gestión requerida (pasos 1, 2, 3 y 4), la segunda la construcción de la implementación de cubos precisos o cuantitativos en un DW, semejante a los explicado en el nivel 1 del punto 3.2 (pasos 5, 6, 7 y 8) y la tercera, la definición de las componentes imprecisas con etiquetas lingüísticas que extenderán el DW a datos cualitativos o datos difuso o Fuzzy Data, semejante al nivel 2 y 3 del punto 3.2 (pasos 9, 10 y 11). La Figura 7 muestra la MFDW, y continuación explicamos cada uno de sus pasos.

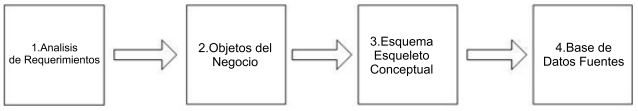
- **Paso 1 Análisis de Requerimientos:** En esta etapa se procede a recolectar los requerimientos del usuario, para la generación de la información que se requiere para la gestión de la organización, por lo general asociados a indicadores de análisis de gestión.
- **Paso 2 Objetos del Negocio:** Considerando los requerimientos obtenidos en la etapa anterior, se procede a definir cuales serán los objetos del negocio, por lo general a partir de indicadores de gestión.
- **Paso 3 Esquema Conceptual:** En esta etapa se procede distinguir los datos u objetos que se requieren para satisfacer los indicadores de gestión del paso anterior. Estos datos deben ser dimensiones (con o sin jerarquías) o medidas generando las relaciones dimensionales obtenidas con los objetos del negocio.
- **Paso 4 Bases de Datos Fuentes:** En esta etapa se procede a analizar la base de datos fuentes, que debe poder permitir extraer los datos para las relaciones existentes entre las dimensiones y medidas del paso anterior.
- **Paso 5 Correspondencia entre Esquema Esqueleto Conceptual y Bases de Datos**

Fuentes: En esta etapa se procede a ubicar las correspondencias entre el esqueleto esquema conceptual (paso 3) y la base de datos fuentes (paso 4). Se verifican que los datos existan y se ubican las fuentes de los objetos conceptuales. Por lo general los datos que están en la base de datos fuentes permiten la extracción de las relaciones de dimensiones, de no ser así, se puede desechar el indicador o bien incorporar el dato en la fuente. Debe quedar claro que los datos son siempre extraídos de la fuentes, no creados en la relación dimensional.

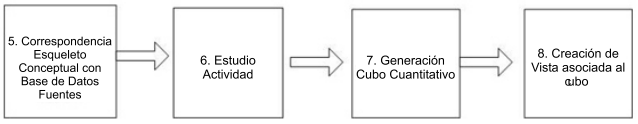
- **Paso 6 Estudio Aditividad:** En esta etapa se procede analizar los tipos de medidas que pueden ser aditivas, semi aditivas o no aditivas.
- Las Flow o Aditivas, conservan la semántica al aplicar la suma como Rollup, realizando DrillUp por cualquier dimensión que se defina. Se refieren a un evento o período y son registrados al final del mismo, el monto de una factura o la cantidad de ventas por día, es un ejemplo de esta medida.
- Las Stock o Semi-Aditivas, conservan la semántica al aplicar la suma como RollUp, al realizar Drill-Up, en todas las dimensiones menos el tiempo. Son registradas en un punto específico del tiempo y se refieren a ese instante. Los saldos de cuentas corrientes o inventarios son ejemplos de este tipo de medidas.
- Las No Aditivas, no conservan la semántica al aplicar la suma como RollUp, al realizar DrillUp, en varias o ninguna dimensión. El precio por Ítem, edades, notas, son ejemplos de este tipo de medida.
- **Paso 7 Generación Cubo Cuantitativo:** En esta etapa se procede a construir los cubos definidos en las etapas anteriores en un diseño lógico relacional de FACT, estrella o copo de nieve. Se pueden usar herramientas que existen en el mercado o bien vistas de cubos (Excel).
- **Paso 8 Creación de Vistas asociadas al cubo:** En esta etapa se procede a generar las vistas o almacén de datos que representan a cada uno de los cubos generados en la etapa anterior. Para la creación de las vistas se puede utilizar el operador CUBE y para el almacén de datos la FACT.
- **Paso 9 Definición de Etiquetas Lingüísticas asociadas a las Medidas:** En esta etapa se procede a generar las etiquetas lingüísticas como tipo de datos Tipo 1 y 2 (definidos en apartado 3.4) para las medidas que tendrán tratamiento difuso, que permiten extender el DW con el uso de la lógica difusa. A estas medidas las llamaremos FuzzyMedida.
- **Paso 10 Definición intervalos de valores para Etiquetas Lingüísticas:** En esta etapa se procede a llenar los intervalos de valores trapezoidales de etiquetas de la FuzzyMedida, definidas en el punto anterior. Debe considerarse que los requerimientos del usuario pueden cambiar según el análisis que se requiera, para una mejor expresión de los datos.
- **Paso 11 Utilización Herramienta FrontEnd:** Con la información suministrada en las etapas anteriores, se está en condiciones de utilizar la herramienta front-end construida para el tratamiento difuso de los cubos que conforman el DataWarehouse difuso.

Figura 7: MFDW conjunto de 11 pasos del Método Fuzzy DataWarehuose.

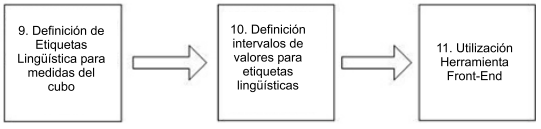
FASE 1: Especificación de requerimientos de datos de gestión.



FASE 2: Implementación de cubos cuantitativos.



FASE 3: Implementación de cubos cuantitativo a-cualitativos.



A continuación, se procederá a implementar con un ejemplo los 11 pasos de MFDW, generando una solución para una empresa que requiere trabajar sus indicadores de gestión con imprecisión. Las etapas siguientes se pueden ejecutar tantas veces como cubos o análisis de medidas sean requeridos por el usuario.

4.1 PASO 1: Análisis de Requerimientos

La empresa para la cual se desarrolla el sistema de gestión, es del ámbito de comercio exterior, y su negocio se aboca a las representaciones internacionales y el comercio exterior, de tal forma que,

sus operaciones cubren el mercado de Chile y el resto del Cono Sur de América. Desde el año 2001, la empresa cuenta con un sistema informático que le permite llevar el registro de todas las transacciones hechas por la empresa de los distintos clientes y proveedores, creando una base de datos operacional. El sistama, tiene más de 6 años de uso, si bien, la operatoria del registro de ventas esta solucionado, se desea conocer información a nivel de totales de ventas para los distintos clientes y proveedores en un determinado espacio de tiempo. Para ello se sugiere construir, una solución MOLAP que genere cubos en un almacén de datos extendiendo sus medidas con la función FuzzyMedida y permita la gestión de la información, mediante indicadores de análisis en un tiempo determinado. La información se almacena en un cubo generado por las dimensiones de clientes, proveedores y el tiempo (jerarquizado en año y luego en meses) y la medida a considerar, es el monto facturado por un cierto período.

4.2 PASO 2: Objetos del negocio

El ámbito de negocio de nuestro caso de estudio, son las representaciones internacionales y el comercio exterior, como se comento anteriormente, se cuenta con una base de datos operacional de más de seis años en ejecución. En el apartado 2.1 se explican dos componentes básicas de los procesos MOLAP, dimensión y medidas, y en este paso se describen, los elementos que se requieren trabajar, para un determinado indicador de gestión. En relación a estas componentes, la Tabla 1, muestra los objetos, su descripción y representación asociada a medidas y dimensiones, elementos indispensables para la generación de cubos y satisfacer el siguiente indicador de gestión.

Indicador: “totales de ventas para los distintos clientes y proveedores en un determinado espacio de tiempo”.

Tabla 1: Objetos del Negocio para el indicador de gestión.

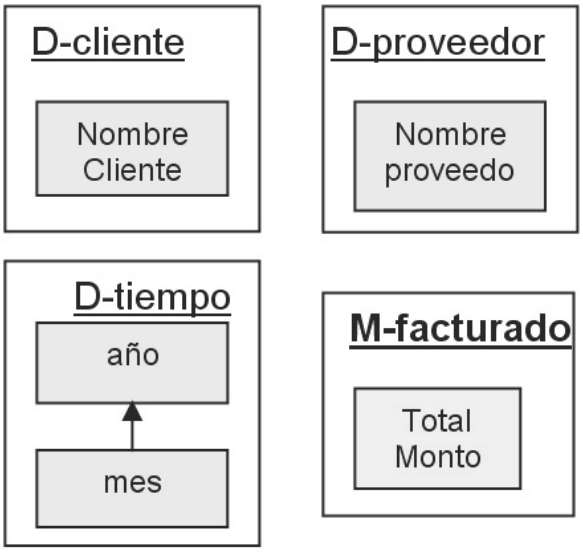
Objeto	Descripción	Medida	Dimensión
Cliente	Cliente que solicita la importación de productos		X
Proveedor	Proveedor que suministra el producto		X
Tiempo	Fecha en que se registra la transacción		X
Valor	Total facturado por la transacción entre un cliente y proveedor	X	

La ”x” en la Tabla 1, indica que ese objeto es requerido como una componente de dimensión o medida.

4.3 PASO 3: Esquema esqueleto conceptual

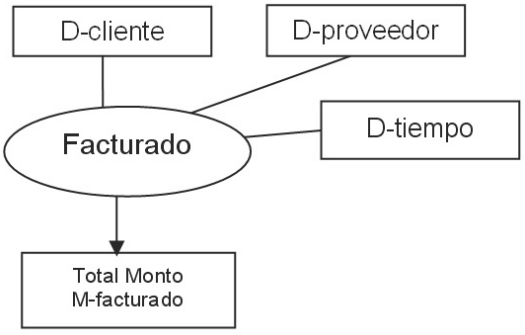
En este paso se deben incorporar los datos de cada una de las dimensiones y medidas que se implementarán para satisfacer el indicador o los indicadores de gestión de la empresa u organización. La Figura 8, muestra el modelado de este tipo de requerimientos utilizando la herramienta CMDM, explicado en el apartado 2.2.

Figura 8: CMDM para Dimensiones y Medidas de indicador total facturado.



Note, que para la dimensión D-tiempo se ha creado una jerarquía de año y mes, en cambio, para las dimensiones D-cliente y D-proveedor no fue necesario. Una vez analizada las dimensiones y medidas se debe generar la relación dimensión que une las dimensiones y medidas. La Figura 9, es una forma abstracta del posible cubo que satisface el indicador solicitado en el paso 2.

Figura 9: CMDM para la Relación facturado del indicador solicitado y posterior cubo.



Se debe recordar que este tipo de relación debe ser generada por tantos cubos como se requieran, según los requerimientos de usuario para el sistema de gestión.

4.4 PASO 4: Datos fuentes

La base de datos fuentes está en SQL Server 2000, y para facilitar la extracción se creó un procedimiento, que permite traspasar los datos de las tablas Proveedor, Cliente y Facturas, por períodos determinados de un mes según la especificación de requerimientos, en este proceso, los datos son limpiados y chequeados. A continuación mostramos las tablas de la base de datos fuentes.

Tabla Proveedores *(id_preveedor, des_proveedor)*
Tabla Cliente *(id_cliente, des_cliente)*
Tabla Facturas *(id_cliente, id_proveedor, fecha_factura, Monto_factura)*

4.5 PASO 5: Correspondencia esquema esqueleto conceptual con base de datos fuentes

La importancia de este paso es la correspondencia entre la relación facturado, presentado en la Figura 9 del paso 3 y los datos de las tablas mostradas en el paso 4. El único dato que es propio de esta etapa es la dimensión tiempo, ya que se refiera al tiempo de extracción de la información (Figura 9, D-tiempo). La Tabla 2 muestra esta correspondencia.

Tabla 2: Correspondencias entre relación (dimensiones y medidas) y tablas fuentes.

Dimensión/ Medida	Datos de las Tablas Fuentes
D-Cliente D-Proveedor D-Tiempo M-Facturado	*id_cliente, des_cliente.* *id_preveedor, des_proveedo.* *mes, año.* *id_cliente, id_preveedor, Monto_factura (operación sum).*

4.6 PASO 6: Estudio de aditividad

En este caso la medida (Monto_factura con operación sum) es del tipo Flor o aditiva. Donde la medida M-facturado, es obtenida por la siguiente fórmula: suma del monto de factura por proveedor y cliente en un tiempo de un mes.

4.7 PASO 7: Creación del cubo cuantitativo

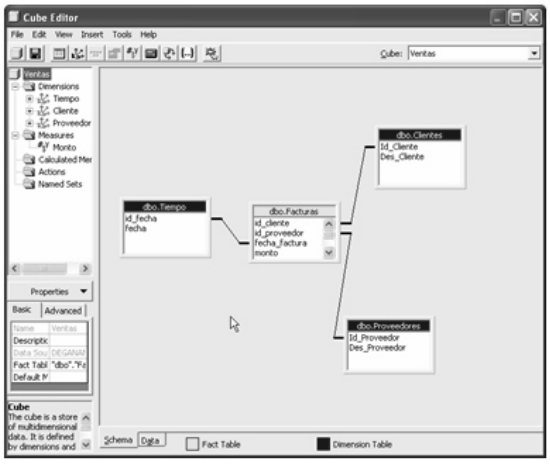
Para este caso, en la construcción del cubo se utilizó la herramienta de Analysis Manager, de Microsoft SQL Server 2000. Su producto final es un cubo, y se consideró para analizar sólo información precisa. Los pasos del uso de la herramienta son:

1.En primer lugar se debe entrar al Analysis Manager, que esta dentro de la opción Analysis Services, que esta dentro de la opción SQL Server.
2.Una vez dentro del Analysis Manager, se procederá a crear una nueva base de datos que contenga el cubo Ventas (clic derecho sobre el nombre de servidor y elegir la opción New Database).
3.Posteriormente, se deberá ingresar el nombre de la base de datos. En este caso se llamara Tesis. Creada la base de datos, hay que indicar cual será el origen de los datos (dentro de la base Tesis, se debe hacer clic derecho sobre la opción Data Sources).
4.Se debe especificar la fuente de origen de datos. Una vez especificada la fuente de origen de datos, se puede proceder a la creación del cubo (se debe hacer clic derecho sobre la opción Cubes/New Cube, y elegir el asistente, luego se debe hacer clic en la opción Next).
5.Luego se pide seleccionar la tabla de hechos, tabla en donde se encuentra la medida, en nuestro ejemplo, es la tabla facturas. Seleccionada la tabla de hechos, hay que elegir el campo que representa la medida del cubo, en nuestro ejemplo, será la columna monto.
6.Posteriormente, se debe crear las dimensiones en la pantalla siguiente, se debe hacer clic en la opción New Dimension. En primer lugar crearemos la dimensión tiempo. En esta dimensión se debe elegir el modelo estrella.
7.A continuación se debe seleccionar la tabla que se ocupará como dimensión de tiempo, en nuestro ejemplo, será la tabla Tiempo, y luego continuar. El asistente solicita el tipo de dimensión, se debe seleccionar la opción Time dimension, automáticamente el asistente presenta el campo o los campos que sean de tipo fecha. Seleccionado el campo se debe continuar. Después se pide seleccionar el tipo de niveles que tendrá la nueva dimensión. Se elige la opción Year, Month, La dimensión tiempo estará jerarquizada por el mes y año. Finalmente se debe continuar hasta donde se solicita el nombre de la nueva dimensión Tiempo.
8.Posteriormente se crean las dimensiones Cliente y Proveedores, ambas en esquema estrella. Finalizada la creación de dimensiones, se procede a ingresar el nombre del cubo Ventas.

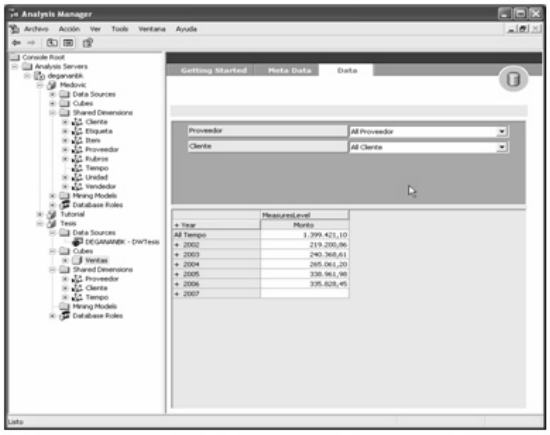
En la Figura 10, se pueden apreciar las dimensiones y medidas del cubo Ventas. Finalizado el cubo, se muestra la estructura de tablas del cubo Ventas. Al salir de esta pantalla se procesa el cubo y finalmente se muestran sus resultados.

Figura 10: a) Cubo Venta generado en Análisis Server en esquema estrella, b) Vista del cubo.

a)



b)



Note que se puede llegar hasta este paso de la MFDW, si el usuario requiere de consultas al cubo que le entregan información cuantitativa o precisa. Pero el cubo generado no tiene la capacidad de responder preguntas tales como: ¿Cuáles fueron los meses con buenas ventas el año 2005?, o ¿Cuales fueron los proveedores con malas ventas en Enero 2005?, que es la problemática que resolvemos a continuación

4.8 PASO 8: CREACIÓN DE VISTAS ASOCIADAS AL CUBO

Otra forma de crear vistas del cubo, es utilizando instrucciones SQL CUBE mostradas en el apartado 3.3, en un uso de operaciones ROLAP. Se puede usar el mismo cubo resultante del paso 7, la diferencia es que aquí se pueden realizar tantas consultas como se requiere, de mayor complejidad y no dependen de la herramienta CASE que tenga el administrador OLAP. La siguiente sentencia muestra las instrucciones generadas para este caso.

```
Create View CuboVentas As Select
Case When (Grouping(Id_Cliente) = 1) Then '-1'
Else Isnull(Id_Cliente, 'Unknown') End Id_Cliente,
Case When (Grouping(Id_Proveedor) = 1) Then '-
1' Else Isnull(Id_Proveedor, 'Unknown') End As
Id_Proveedor,
Case When (Grouping(Anio) = 1) Then -1 Else
Isnull(Anio, 'Unknown')  End As Anio,
Case When (Grouping(Mes) = 1) Then -1 Else
Isnull(Mes, 'Unknown')  End As Mes,
Sum(Monto) Monto
From Facturas
Group By Id_Cliente,Id_Proveedor,Anio,Mes
With Cube
```

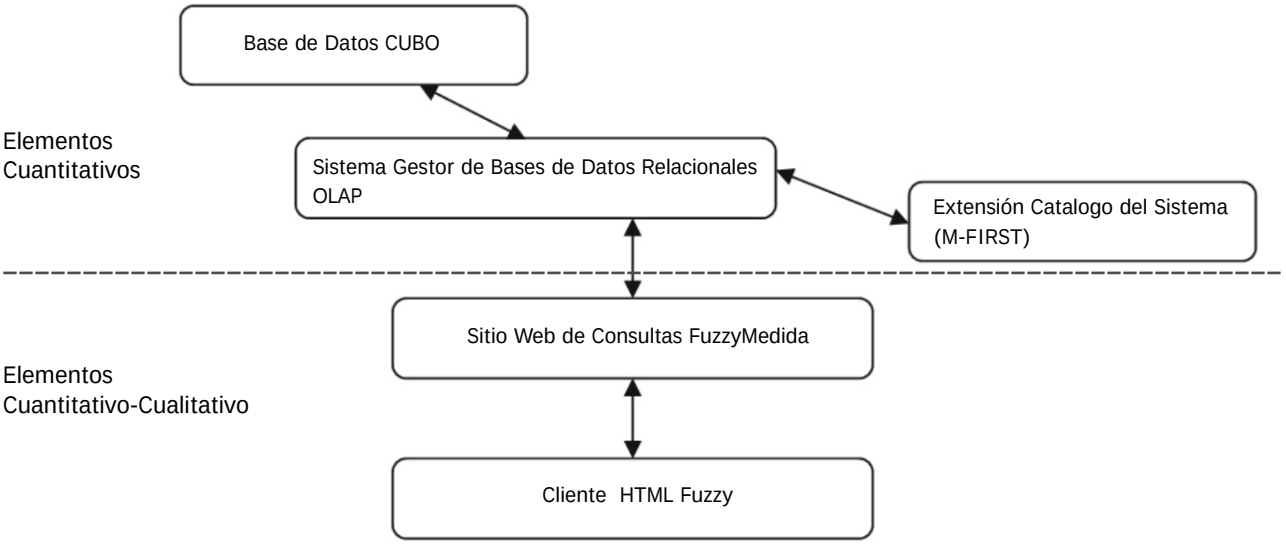
Para los valores *Null* que se generan y que corresponden al valor *All*, se les cambia el *Null* por el -1 a través del uso de la función *grouping*. Esta vista permite trabajar con el cubo a nivel numérico sin tratamiento difuso.

Este paso es opcional ya que vistas del cubo también se obtuvieron en el paso 7.

4.9 PASO 9: Definición de etiquetas lingüísticas asociadas a las medidas (FUZZYMEDIDA)

Para los pasos de aquí en adelante se utiliza la extensión del cubo generado en el Paso 7, la arquitectura mostrada en la Figura 11 representa una extensión de un motor de bases de datos relacional que permita procesos OLAP y consultas sobre sus medidas con etiquetas lingüísticas usando la transformación FuzzyMedida. Para ello se extiende el catálogo usando la FIRST [5] presentada en el apartado 3.4, y se construye una herramienta Front-End, que permite manipular el cubo con el operador *CUBE*.

Figura 11: Arquitectura del sitio para consultas difusas.



A continuación damos una breve descripción de las componentes de la arquitectura propuesta.

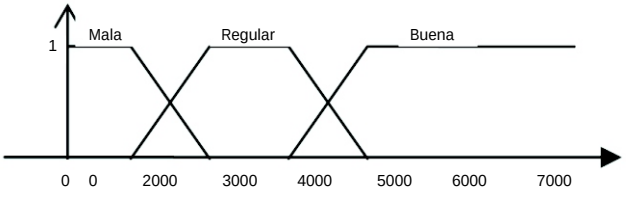
Base de Datos Cubo: Almacena en formato relacional toda la información que sea de utilidad para el tratamiento de los cubos, en forma de estrella o copo de nieve. Es igual que cualquier otra base de datos, pero permitirá el almacenamiento de información difusa.

Extensión Catalogo del Sistema: Es un módulo donde se procede a extender el catálogo del sistema del SGBD, agregando tablas (Véase Figura 6) para el tratamiento difuso que tienen como base la FIRST [5] creando M-FIRST, e incorporando dos funciones (FLabel y FGrado) para las componentes de la FuzzyMedida.
SGBDR: Todas las operaciones que se desean aplicar a FuzzyMedida, se traducirán a peticiones al SGBDR anfitrión.
Sitio Web de Consultas Difusas: Corresponde a la herramienta Front-End, construida que permite generar consultas difusa a los cubos generados usando la instrucción CUBE.
Cliente HTML: Cliente o usuario final que se construyó en ASP, y se uso de herramienta Fron-End por medio de un navegador.

4.9.1 Implementación extensión

La extensión FuzzyMedida, para este caso es el dato Monto incluido en el cubo Ventas. La medida Monto, es un atributo Tipo 2 con referencial ordenado (explicado en apartado 3.4) y la función asociada a dicha medida será la de una distribución de posibilidad trapezoidal mostrada en la Figura 12. Que se implementan de la siguiente forma.

Figura 12: Función trapezoidal para la medida Monto de factura.



Definición de Etiquetas Lingüísticas e intervalos de valores para medidas del Cubo:

A la medida Monto, según la especificación de requerimiento le corresponderán tres etiquetas lingüísticas Mala, Regular y Buena de la Figura 12, por tanto se debe completar las siguientes tablas FIRST (Véase Figura 6): FUZZY_COL_LIST (FCL), FUZZY_LABEL_DEF (FLD) y FUZZY_OBJECT_LIST (FOL) según [5].

En primer lugar la información debe estar registrada en la tabla FCL. Luego el registro de la medida quedaría como se muestra en la Tabla 3. De haber más medidas con tratamiento FuzzyMedida, bastaría con agregarlas como un fila más de la tabla FCL, cuyo acceso es con la clave primaria OBJ#COL#.

Tabla 3: Tabla FCL para la medida Monto facturado.

OBJ#	COL#	F_TYPE	LEN	COM
Facturas	Monto Factura	2	2	Medida Monto

Posteriormente hay que registrar las etiquetas lingüísticas en la tabla FOL mostrada en la Tabla 4, para la medida Monto facturado registrada en FCL, según Figura 12. De querer definir mas etiquetas lingüísticas, bastará con ingresarlas otras filas de la tabla FOL, cuyo acceso es con la clave primaria OBJ#COL#FUZZY_ID.

Tabla 4: Tabla FOL que define las etiquetas lingüísticas de Monto factura.

OBJ#	COL#	FUZZY _ID	FUZZY _NAME	FUZZY _TYPE
Factu	Monto F.	0	Mala	0
Factu	Monto F.	1	Regular	0
Factu	Monto F.	2	Buena	0

Luego para cada etiqueta lingüística, se debe definir los valores de la función trapezoidal (Figura 12), asociada a cada etiqueta. Esto se hace en la tabla FLD como se muestra en la Tabla 5, cuyo acceso es con la clave primaria OBJ#COL#FUZZY_ID.

Tabla 5: Tabla FLD para los datos de cada etiqueta de Monto factura.

OBJ#	COL#	FUZZY _ID	ALFA	BETA	GAMMA	DELTA
Factura	Monto	0	0	0	1000	2000
Factura	Monto	1	1000	2000	3000	4000
Factura	Monto	2	3000	4000	6000	7000

Cabe destacar que estas tablas se levantan junto con la base de datos, sin embargo esta representación, si bien sirve para poder consultar de manera difusa información detallada de las ventas, no sirve para almacenar información difusa en el cubo ventas. Para lo que se crea una nueva tabla construida para dar soporte a consultas difusas sobre un cubo.

4.9 Extensión catalogo del sistema

A la FIRST presentada en la Figura 6, se le agrega una nueva tabla llamada FLM la que se asocia a la tabla FDL. FLM, permitirá almacenar las componentes, dimensiones y medidas del cubo, para poder ejecutar operaciones difusas. Es este, la que describimos a continuación.

FUZZY_LABEL_MULTIDIMENSIONAL (FLM): La tabla FLD, no fue diseñada para realizar consultas difusas a un Cubo donde la medida intersecta a una o varias dimensiones. Para llevar a cabo dicho tratamiento, se crea una nueva tabla que se adosa FLM a la FIRST, creando la M-FIRST y su estructura es la siguiente:

- OBJ#: Almacena el número de objeto de la tabla que tiene un atributo difuso.
- COL#: Almacena el número de columna dentro de la tabla que admitirá un tratamiento difuso. En este caso corresponde a la medida
- FUZZY_ID: Identificador del objeto difuso asociado a la tabla FOL
- D_1 : Primera dimensión del cubo
- N_1 : N° de nivel de la primera dimensión
- D_2 : Segunda dimensión del cubo
- N_2 : N° de nivel de la segunda dimensión
- ....
- D_n : Dimensión n del cubo
- Nn : N° de nivel de la dimensión n
- ALFA, BETA, GAMMA Y DELTA: Definen una distribución de posibilidad trapezoidal, para la medida y dimensiones especificadas.

En el caso de implementarse para un Sistema de Datawarehouse, la dimensión Dn, estará dada por el cubo que presente la mayor cantidad de dimensiones. Por Ejemplo: Si hay dos cubos y uno posee tres dimensiones y otro 5, la tabla FLM tendrá desde D1 hasta D5 columnas asociadas a esta tabla. Esta tabla permite tratamiento difuso para cubos, cuyas medidas sean datos de Tipo 1 o 2. Con esta tabla se puede dar solución a las operaciones Slice, Dice, Drill-up, Drill-down, sobre un cubo. Las columnas de dimensiones pueden tomar 3 valores distintos 0, -1 o un valor específico para una dimensión. -1, significa que se esta consultando por el valor All, la dimensión 0, significa que no se esta consultando por el valor All, sino por el valor asociado a la dimensión. Cualquier otro valor, corresponde a un atributo específico para la dimensión, que tenga asociado tratamiento difuso. El acceso a M-FIRST es con la clave primaria OBJ#COL#FUZZY_ID.

4.10 PASO 10: Definición de intervalos de valores para etiquetas lingüísticas

En primer lugar se deben registrar los trapecios asociados al cubo difuso, tal como lo muestra la Tabla 6. En la tabla FLM se representan todas las combinaciones de las dimensiones del cubo con una medida, a cada combinación se le asigna el trapecio de la etiqueta lingüística.

Tabla 6: Tabla de valores de FuzzyMedida en Fuzzy_Label_Multidimensional para medida Monto.

OBJ#	COL#	FUZZY_ID	D1	N1	D2	N2	D3	N3	alfa	Beta	Gamma	delta
Factura	Monto	Mala	0	1	0	1	0	1	0	0	3000	5000
Factura	Monto	Buena	0	1	0	1	0	1	3000	5000	7000	10000
Factura	Monto	Regular	0	1	0	1	0	1	7000	10000	100000	100001
Factura	Monto	Mala	0	1	0	1	all	1	0	0	7000	11000
Factura	Monto	Buena	0	1	0	1	all	1	7000	11000	12000	15000
Factura	Monto	Regular	0	1	0	1	all	1	12000	15000	1000000	1000001
Factura	Monto	Mala	0	1	all	1	0	1	0	0	10000	13000
Factura	Monto	Buena	0	1	all	1	0	1	10000	13000	15000	20000
Factura	Monto	Regular	0	1	all	1	0	1	15000	20000	100000	100001
Factura	Monto	Mala	0	1	all	1	all	1	0	0	5000	10000
Factura	Monto	Buena	0	1	all	1	all	1	5000	10000	12000	15000
Factura	Monto	Regular	0	1	all	1	all	1	12000	15000	100000	100001
Factura	Monto	Mala	all	1	0	1	0	1	0	0	5000	12000
Factura	Monto	Buena	all	1	0	1	0	1	5000	10000	13000	15000
Factura	Monto	Regular	all	1	0	1	0	1	13000	15000	100000	100001
Factura	Monto	Mala	all	1	0	1	all	1	0	0	7000	10000
Factura	Monto	Buena	all	1	0	1	all	1	7000	10000	12000	15000
Factura	Monto	Regular	all	1	0	1	all	1	12000	15000	200000	200001
Factura	Monto	Mala	all	1	all	1	0	1	0	0	70000	110000
Factura	Monto	Buena	all	1	all	1	0	1	70000	110000	120000	140000
Factura	Monto	Regular0	all	1	all	1	0	1	120000	140000	200000	200001

La combinación de la medida con sus dimensiones es una información suministrada por el usuario experto en el dominio del problema. En nuestro caso de ejemplo, se tiene la medida Monto venta y sus dimensiones cliente (D1), proveedor (D2) y tiempo (D3) que se compone de una jerarquía de año y mes. Teniendo clara esta información se procede al llenado de la FLM. Sucesivamente se procederá hacer el llenado de las dimensiones clientes, proveedor y tiempo, en esta última se trabajó con el primer nivel del año. La tabla 6 muestra resultados de esta extensión.

4.10.1 Extensión del SGBDR

Para el tratamiento difuso del cubo generado en la sección anterior, se crearon dos funciones asociadas la FuzzyMedida y la tabla FLM :

FLabel : Esta función recibe como parámetros el valor de la columna de medida, las dimensiones con su respectivo nivel de dimensión, el valor de la medida, y un indicador que especifica si se desea la etiqueta lingüística a la que más se acerca o menos se acerca al valor dado. Como la medida es un dato Tipo 2, puede pertenecer a n etiquetas lingüísticas por la particularidad de la función trapezoidal. Lo que variará es el grado de pertenencia a una u otra. Por ejemplo en la función trapecio de la Figura 12, el valor 1001 le corresponde la etiqueta lingüística Mala, pero también pertenece en menor medida a una etiqueta lingüística Regular. Esta función retorna un string con las etiquetas lingüísticas asociadas.

FGrado: Esta función recibe como parámetros, el valor de la columna de la medida y dimensiones con su respectivo nivel de dimensión. El valor especifica la medida, y un indicador si se desea, a la etiqueta lingüística a la que más se acerca o menos se acerca al valor dado. Esta función retorna un número que corresponde al grado de pertenencia del valor dado para la etiqueta lingüística, a la cual pertenece, en mayor o menor medida.

4.11 Construcción herramienta FRONT-END

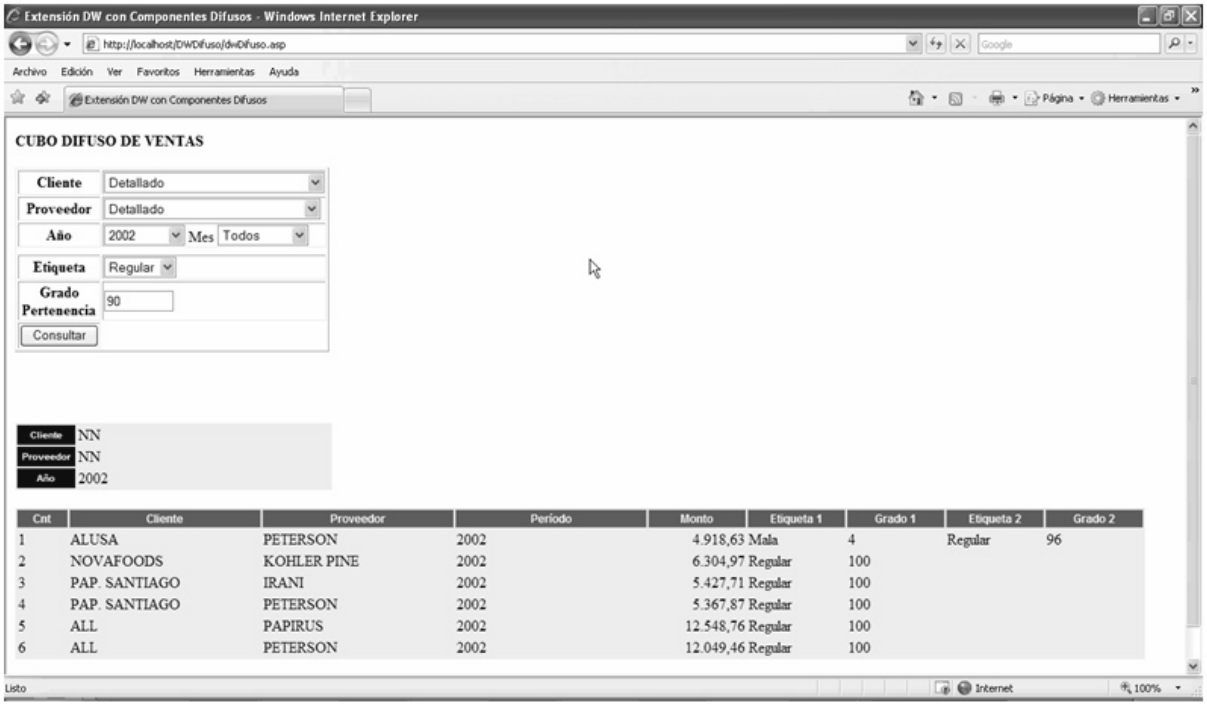
Para la generación de la herramienta Front-End, se construyó una página Web en ASP. Esta página tiene como parámetros de entradas las dimensiones del cubo (Cliente, Proveedor y tiempo compuesto por Año y Mes), y dos atributos más que permiten dar el tratamiento difuso al cubo, considerando el atributo, etiqueta lingüística y grado de pertenencia, donde la etiqueta lingüística corresponde a las definidas en la tabla FOL, para este caso Mala, Regular o Buena. Esta herramienta permite hacer consultas al cubo de manera normal, dejando el valor de la etiqueta en Ninguno. Al realizar esto, se despliega el valor de la medida junto con la etiqueta lingüística asociada a dicha medida. Estas etiquetas lingüísticas están definidas por los valores que fueron registrados en la tabla Fuzzy_Label_Multidimensional.

Algunos ejemplos de consultas son: Consultar por el comportamiento de un proveedor en relación a todos los clientes a los que les exportó productos en un año en particular. Consultar por el grado de pertenencia de una

medida con alguna etiqueta en particular. Consultar por las ventas Regular, del año 2002, con un grado de pertenencia a dicha etiqueta mayo o igual a 90%. Una muestra de consultas se muestran en la Figura 13.

También, con esta aplicación se pueden desarrollar operaciones correspondientes a Drill-up sobre la dimensión tiempo, Drill-down sobre la dimensión tiempo, es decir, mostrando la información por año y mes para los años 2004 y 2005. Consultar por los proveedores con ventas Buenas, para el año 2003.

Figura 13: Muestra del Resultado para un año con grados de pertenencia.



5. CONCLUSIONES

El trabajo muestra un conjunto 11 de pasos que da un Método Fuzzy para implementar DW, llamado MFDW, y aplicado a un caso práctico de implementación. Los principales aportes de MFDW son:

a)FuzzyMedida que es una extensión del cubo tradicional que retorna información cualitativa, para medidas asociado a etiquetas lingüísticas.
b)FML extiende la FIRST propuesta por (Galindo, 1999) generando la M-FIRST que incorpora una nueva tabla Fuzzy_Label_Multidimensional, que se acopla al modelo ya propuesto y permite el tratamiento de cubos multidimencionales.
c)FLabel y FGrado funciones que permiten asociar grados con las etiquetas lingüísticas y viceversa implementadas en el gestor de bases de datos.
d)Herramienta Fron End para Web que permite consultar una FuzzyMedida en un Fuzzy DW para cualquier BDMS con OLAP utilizando la instrucción CUBE.
e)La herramienta presentada permite trabajar con operaciones Slice, Dice, Drill-up, Drill-down entre otras para cubo.

La forma de extender las fases del apartado 4, es bastante sencilla en relación a los trabajos presentados en el apartado estado del arte, ya que al utilizar la función trapezoidal, cuya definición de los intervalos deben estar claros para el usuario experto en el dominio del problema, son de fácil ingreso, y para una misma medida se pueden utilizar tantas etiquetas lingüísticas como sea necesario.

La extensión se hizo utilizando el operador CUBE, el cual forma parte desde el 2003, de los operadores estándar en el SQL. La extensión permite trabajar tanto con modelo estrella como copo de nieve del cubo. Se hizo una extensión del SQL Server, por medio de dos funciones que permiten tratamiento difuso a un cubo en particular (FLabel y FGrado).

La aplicación de nuestro caso, fue en una organización

en la cual los usuarios quedaron satisfechos con los resultados de las etiquetas lingüísticas implementadas en su sistema de gestión para la toma de decisiones.

Trabajos futuros: implementar las etiquetas lingüísticas para: Fuzzy dimensión, Fuzzy cuantificador, Fuzzy comparador. Además, perfeccionar la herramienta front-end, para otros motores de bases de datos.

Agradecimientos: Al proyecto interno "Extensión de Almacenes de Datos (DW) en el Diseño y Consultas con Incertidumbre Usando Lógica Difusa" fecha de inicio 1-1-2006, fecha de termino 31-12-2007. adjudicado en la Universidad Católica del Maule, número 81201(2006-2007).

6. Bibliografía

[1]Cheung Pui Ling Pauline, Lau Wai kay Ricky, Lee Tak Wn Angus, Tsoi Chin Ching Lancelot and Yip Keung Frank, "Data Warehousing and OLAP". CiteSeer.IST homepage, 1997. http://www.cs.ust.hk/faculty/dimitris/COMP530/OLAP.pdf.

[2]Carpani Fernando, "CMDM: Un Modelo Conceptual para la Especificación de Bases Multidimensionales", Tesis de Maestrita, Universidad de la República, Uruguay, 2000.

[3]Delgado Garrón Alberto, "Descubre Microsoft SQL Server 7", editorial Prentice Hall, 1999.

[4]Figueroa González Jessica, "Implementación de un Almacén de Datos para una base de datos DB2 usando instrucciones SQL: Una solución ROLAP", Tesis Ingeniería Civil Informática, Universidad Católica del Maule, Chile, 2007.

[5]Galindo José, "Tratamiento de imprecisión en Bases de Datos Relaciónales: Extensión del modelo y adaptación a los SGBD Actuales", Tesis Doctoral, Universidad de Granada España, 1999.

[6]Galindo Jose, Urrutia Angélica, Piattini Mario, Fuzzy Databases: Modeling, Design and Implementation, Editorial Group Idea Publishing, USA, 2006.

[7]Gavin Powel, Beginning Database Desing, Wiley Publishing, Inc. USA, 2006.

[8]Golfarelli, M. Maio, D. Rizzi, S., "Conceptual Desgin of Data Warehouses from E/R Schemes", International Conference on System Sciencie, Hawai, IEEE, 1999.

[9]Herder Olaf, "A Design Methodology for Data Warehouses", Oldenburg Research and Development Institute for Computer Science Tools and Systems (OFFIS), Germany, 2000.

[10]Inmon, W., "Building the Data Warehouse". John Wiley & Sons, Inc. 1996.

[11]Kimball R., "The DataWarehouse ToolKit". Jhon Wiley & Son, Inc, 1996.

[12]Kumar Pavan, Krishna Radha, Kumar Supriya, "Fuzzy OLAP Cube for Qualitative" Institute for Development and Research in Banking Technology, IDRBT-2005

[13]Ling Fen, Tharam Dillon, "Using Fuzzy Linguistic Representations to Provide Explanatory Semantcis for Data Warehoses", IEEE, Vol. 15 N°1, Enero-Febrero 2003.

[14]Melton Jim, "(ISO-ANSI Working Draft) Foundation (SQL/Foundation)", ISO/IEC 9075-2:2003 (E), United States of America (ANSI), 2003.

[15]Microsoft, http://technet.microsoft.com/es-es/library/ms175939.aspx, artículo sobre el uso del operador CUBE, 2007.

[16]Peralta Verónika, "Diseño Lógico de Data Warehouses a partir de Esquemas Conceptuales Multidimensionales", Tesis de Maestría, Universidad de la Republica Uruguay, Uruguay, 2001.

[17]Salas Olivares Yoselin, "Extensión del Diseño e Implementación de un Data Warehouse, para tratamiento de datos con Lógica Difusa, aplicado a enfermedades cardiovasculares", Seminario de Titulo, Universidad Católica del Maule, Chile, 2006.

[18]Zadeh, Lord A. Fuzzy Sets. Information Control, 8:338-353, 1965.